Research and optimization of OpenStack virtual machine resource scheduling technology

Jun Zhao

***Abstract*—**Cloud computing is a new business model of information technology. It services clients with the visualized hardware infrastructure resources in dynamic and flexible ways. The suppliers increase the input for infrastructure because of the high popularity of the service, leading to increased energy consumption and carbon emission. To reduce energy consumption and improve resource utilization rate, it is urgent to select proper physical resource distribution and virtual machine dispatching means. This study analyzed OpenStack cloud platform, proposed multi-objective ant colony optimization algorithm for multi-objective optimization issue and redesigned the resource scheduling model of server to efficiently satisfy user requirements. Moreover other algorithms including first fit algorithm, least residue algorithm and conventional ant colony algorithm were simulated. The experimental parameter conditions included 100 physical nodes, a physical host (CPU computing power array: (1000, 1500, 2000); internal storage: 4 GB; storage: 1 TB; bandwidth: 1 Gpbs), 200 virtual machines (CPU computing array: (200,400,600,800), computer frequency: (0.2GHz, 0.4GHz, 0.6GHz, 0.8GHz); internal storage: 1GB; storage: 400 GB; bandwidth: 250 Mbps), 150 ~ 200 W power consumption, and concurrent execution of 200 tasks. The four algorithms were simulated under the above conditions. The service level agreement (SLA) violation rate, resource balance and placement superiority of the four algorithms were analyzed. It was found that first fit algorithm was moderate in service quality and energy consumption, but the randomness was high; least residue algorithm was excellent in service quality, but the energy consumption was high; the conventional ant colony algorithm and the improved ant colony algorithm had excellent service quality and low energy consumption. The analysis of the placement superiority of the four algorithms suggested that multi-objective ant colony optimization algorithm had good balance in service quality and energy consumption. In conclusion, the proposed multi-objective ant colony optimization algorithm can reduce energy consumption on the premise of ensuring service quality.

***Keywords*—**Ant colony optimization algorithm, OpenStack virtual machine, resource scheduling technology

I. Introduction

CLOUD computing as a new technology can provide computing resource service based on an easily extended method according to the computation requirements of users [1]. With the development of computer information technology, the requirements of users on cloud computing platform are increasing; hence how to schedule resources as required has been a hot research subject. Chen et al. [2] established an OpenStack basic architecture platform, proposed dynamic resource distribution and energy saving algorithm, realized the algorithm through the live migration of virtual machine and checked whether the algorithm was correct through monitoring system state with power distribution unit and recording power consumption. Yang et al. [3] proposed OpenStack cloud platform based optimized virtual machine migration technology and reduced energy consumption of data center by avoiding ineffective migration through proper threshold and time sequence prediction technology. Xu et al. [12] solved the problem of resource competition in multiple virtual machines operated on the same host by dividing OpenStack users and virtual machines on OpenStack into different levels. Moreover the virtual machines of different levels were given corresponding threshold strategy. Yamato et al. [13] put forward a material resources and logistic virtual management server to realize the Iaas service of OpenStack. The server could reduce the waiting time for image deployment and extraction based on the composite performance of multiple application programming interfaces (API). Nishizawa et al. [16] put forward a high-efficient and reliable restoration method for virtual resources to cope with the situation when physical server crashed. This study analyzed the basic characteristics, technical framework and core technology of OpenStack cloud platform and proposed multi-objective ant colony optimization algorithm for the problems occurring in data migration. The improved ant colony algorithm was more efficient in solving the problem of OpenStack virtual machine resource scheduling. Finally simulation experiment was carried out on several common algorithms on Cloudsim platform. The results demonstrated that multi-objective ant colony optimization algorithm was more energy-saving and had the highest placement superiority; hence it was applicable to OpenStack virtual machine resource scheduling.

Jun Zhao is with Tianjin Polytechnic University, Tianjin, 300387, China (e-mail: junzhao_tj@yeah.net).

II. Procedure for Paper Submission Study of OpenStack Virtual Machine

A. Basic concept of OpenStack

OpenStack is a free software and open source code project which was launched by National Aeronautics and Space Administration and Rackspace and authorized by Apache license [4]. OpenStack has a promising prospect in cloud technology and gradually become the mainstream cloud platform. Openstack has the following characteristics. First is flexible management. Users can explore source code and check

API through OpenStack and add third-party technologies to OpenStack module according to their demands. Second is as the industrial standard. Since many multinational technology companies joined OpenStack, it has been the standard in the whole industry. Moreover, because of the corresponding software compatibility and connection service provided by several cloud computing companies in the industry, data can migrate between different cloud computing servers, which give OpenStack a huge advantage [5]. It can also bring huge convenience. Here the simulation of mass data with MATLAB [18] is taken as an example. The simulation operated on local server will consume 24 hours because of the limitation of lab resources, and no time will be left for other experiments. But if the virtual machine resource of OpenStack is used, several hours can be saved for other experiments [17].

B. OpenStack architecture

OpenStack architecture mainly includes five layers [6].

(1) Resource layer. There are many resources in the layer such as computing, network and storage. Those resources will be sent to the upper-layer users after being integrated by virtualization technology.

(2) Logical layer. The main function of the logical layer is to provide cloud computing with information such as deployment, dispatch, strategy and log.

(3) Display layer. It is mainly responsible for satisfying users' requests, achieving information interaction between different components of OpenStack and sending results to users. Information interaction in this layer is realized through control panel Dashboard. The control panel provides users with operation interface to help them achieve the operation of OpenStack system.

(4) Integration layer. The layer includes the identify information of users and the charge systems of different resources used by users.

(5) Management layer. The layer provides administrator with the API of different components of OpenStack; as a result, administrator can maintain cloud platform regularly.

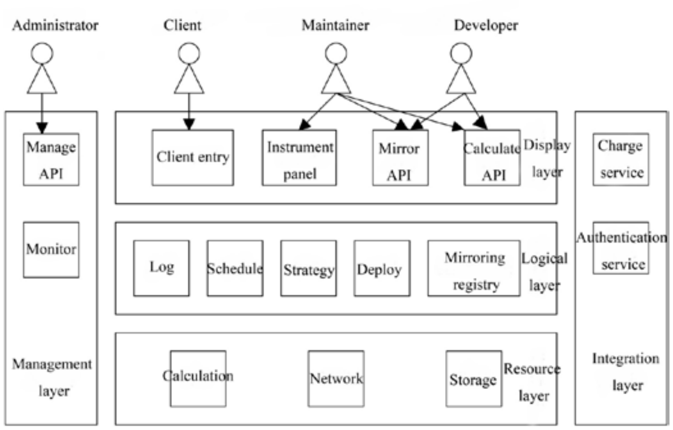

Fig. 1 OpenStack architecture

C. OpenStack components

OpenStack offers services such as computing, storage, network, mirroring, identity authentication and interface management [7].

Any user can install OpenStack cloud platform and corresponding components according to their demands.

(1) Computing service. Computing service component as the core of OpenStack is mainly responsible for managing the life cycle of OpenStack virtual machine.

(2) Network service. Network service can provide virtual network for OpenStack virtual machine to achieve network communication between virtual machines and between virtual machine and physical machine.

(3) Storage service. Storage service includes object storage and block storage. Through object storage service, users can upload and download documents. Through block storage service, users can directly modify the stored document and protect data on virtual machines.

(4) Identity authentication service. Identity authentication service can satisfy the request of identity authentication and divide the authority of users. Users with different identities have different authority of resource visit. Message transmission between different components also needs to be certified to ensure the safety of documents.

(5) Mirroring management service. Mirroring management service can help users efficiently find out the mirroring document of virtual machine and moreover has functions of login and retrieval. Through mirroring management service, users can establish a virtual machine quickly through template and restore virtual machine according to mirror images.

(6) Interface management service. Interface management service provides a Web control interface, through which administrator can manage various services on OpenStack platform more efficiently.

III. METHODS

Currently, there are four algorithms for OpenStack virtual machine resource scheduling. The first algorithm is first fit algorithm [19] which allocates the most suitable virtual machine to host. The method is featured by simple calculation. The randomness of the algorithm is high though its service quality and resource utilization rate are moderate. The second algorithm is least residue algorithm which calculates the optimal solution starting from some solution to a problem [20]. Using least residue algorithm, virtual machines will be dispersed to different physical hosts, which can improve the service quality. But the energy consumption is higher because of the application of many hosts. The third algorithm is conventional ant colony algorithm [21] which finds out the shortest routine by optimizing resource allocation path with information element. The algorithm with positive feedback mechanism can converge in a high speed and moreover has advantages of high service quality and low energy consumption. The last one is multi-objective ant colony optimization algorithm which rearranges the redistribution of virtual machine to solve the placement problem of virtual machine under multiple objectives. Migration of virtual machine manifests as allocating virtual machines to corresponding hosts. It is the improved version of the conventional ant colony algorithm which has not only the advantages of the conventional ant colony algorithm.

This study put forward solving the problem of OpenStack virtual machine resource scheduling using multi-objective ant colony optimization algorithm.

The procedures of multi-objective ant colony optimization algorithm are as follows.

In the initial stage, all the parameters must be initialized, and the concentration value of information element is expressed as β.

As to the iteration section, the request of virtual machine will be sent to all ants, and moreover a new virtual machine is allocated to the current host.

The concentration of information element drives ants to put a virtual machine with optimal value into the current host.

Each movement of ants can induce the changes of local information element concentration.

After all ants find out the new solution, the global information element concentration will change according to the current solution.

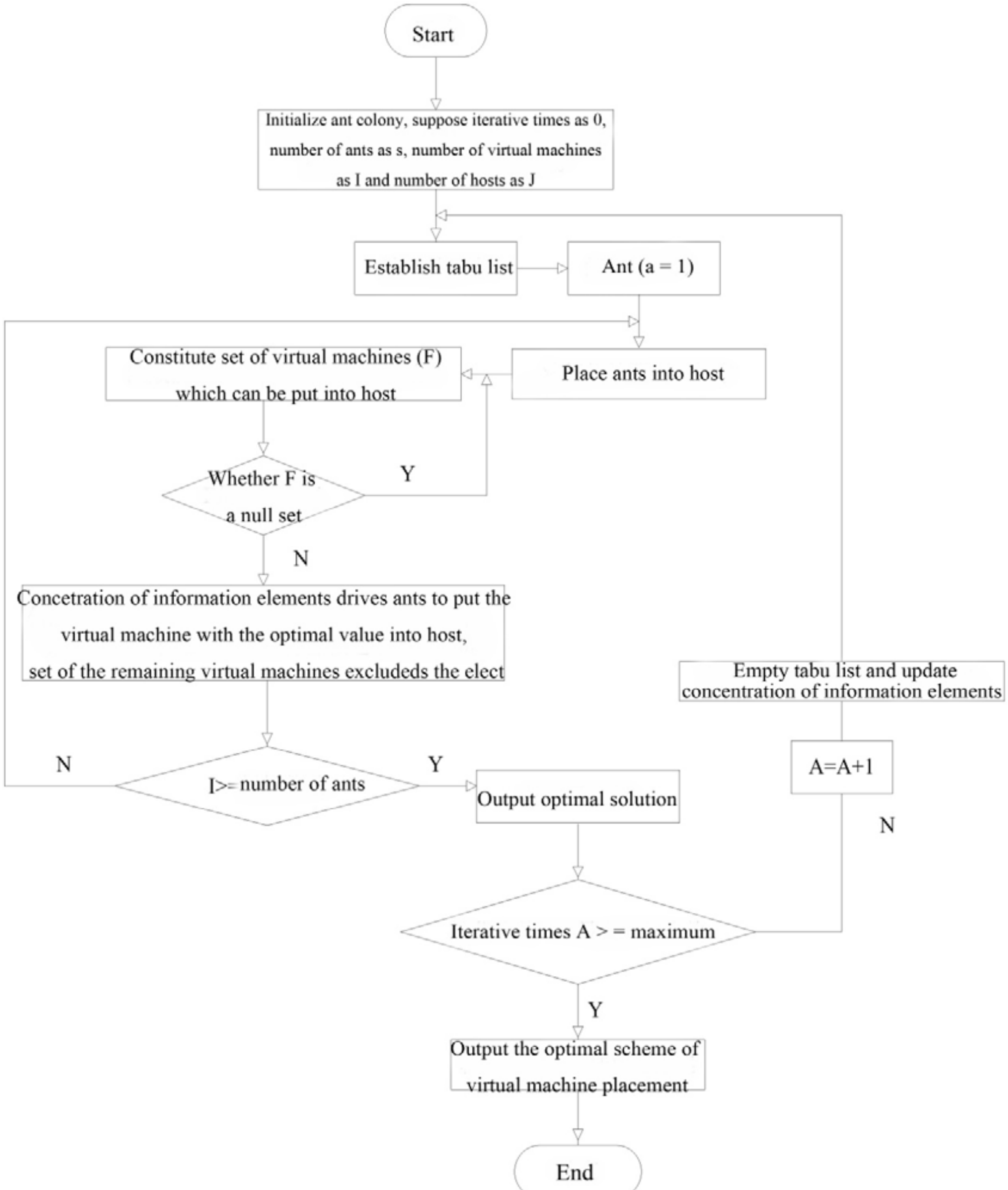

Fig. 2 The flow of multi-objective ant colony optimization algorithm

A. Information elements and heuristic search

Multi-objective ant colony optimization algorithm is realized according to matrix of information element concentration and matrix of heuristic information [8]. The boundary value of information elements almost determines the efficiency of the algorithm; the boundary value of information element is critically important [9]. To solve the problem, this study adopted information element concentration and methods which correlated migration of all virtual machines.

a. Concentration of information elements

The information element concentration of virtual machine m in host n was denoted as $\omega_{m,n}$. In the process of initialization, the computational formula for initial information element concentration was as follows.

$$\omega_0=1/\left[i\bullet\left(\sum_{n=1}^{j}(K_n/K_n^{Max})+M\left(G_0\right)\right)\right], \tag{1}$$

where i stands for the number of virtual machine, G_0 stands for the value calculated by single objective algorithm, $M\left(G_0\right)$ stands for the resource waste condition of host at G_0, $\sum_{n=1}^{j}(K_n/K_n^{Max})$ stands for the power consumption of G_0, and K_n^{Max} stands for the maximum power consumption of host n.

b. Heuristic information search

Heuristic search for information determines the probability when ants placing virtual machines; moreover the optimal solution can be found out by combining heuristic search for information with the concentration of information elements [10]. The expectation degree of allocating virtual machine m to host n was denoted as $\theta_{m,n}$.

Then the objective function for allocating virtual machine m to host n in the first stage could be calculated using the following formula.

$$\theta_{m,n,1}=\frac{1}{\alpha+\sum_{r=1}^{n}(K_r/K_r^{Max})}, \tag{2}$$

where α stands for an extremely small real number, 0.0001, and r stands for a host.

The algorithm of the second stage was similar to that of the first stage.

$$\theta_{m,n,2}=\frac{1}{\alpha+\sum_{r=1}^{n}M_r} \tag{3}$$

The total expectation degree of allocating virtual machine m to host n could be denoted as:

$$\theta_{m,n}=\theta_{m,n,1}+\theta_{m,n,2}. \tag{4}$$

B. Optimal solution algorithm

Ant p allocated virtual machine m according to the following formula.

$$m=\begin{cases}\arg\ \max_{t\in\Omega p(n)}\left\{\tau\times\omega_{t,j}+(1-\tau)\times\theta_{t,j}\right\}, h\le h_0\\ s, otherwise\end{cases}, \tag{5}$$

where τ stands for the parameter of information element concentration controlled by users and h stands for a random numerical value in interval $[0,1]$. If $h>h_0$, then heuristic search started; if $h<h_0$, then the virtual machine released the request of migration. h_0 is a fixed parameter which refers to the relative importance of exploring the next procedure through accumulating knowledge. $\Omega p(n)$ stands for the virtual machine in the set of virtual machine F which could satisfy the request of host n and t stands for a virtual machine which satisfied the request of host n. Hence

$$\Omega_p(n)=\left\{m\in\{1,\ldots,i\}\right\}\left|(\sum_{t=1}^{j}x_{mt}=0)\wedge((\sum_{t=1}^{i}(x_{tn}\times R_{kt})+R_{km})\le T_{kn})\wedge((\sum_{t=1}^{i}(x_{tn}\times R_{jt})+R_{jm})\le T_{jn})\right., \tag{6}$$

where R_{kt} stands for CPU usage of virtual machine t, T_{kn} stands for the maximum CPU usage that could be provided by host n, R_{jt} stands for the memory usage of virtual machine t, T_{jn} stands for the largest memory usage that could be provided by host n, x_{tn} stands for allocating virtual machine t to host n.

Hence the computational formula of probability of allocating virtual machine m to host n was:

$$k_{m,n}^{p}=\begin{cases}\dfrac{\tau\times\omega_{m,n}+(1-\tau)\times\theta_{m,n}}{\sum_{t\in\Omega p(n)}(\tau\times\omega_{m,n}+(1-\tau)\times\theta_{m,n})}, m\in\Omega p\left(n\right)\\ 0, otherwise\end{cases}. \tag{7}$$

C. Updating of information elements

Updating of information elements is also an important part of multi-objective ant colony optimization algorithm; information elements will increase when ants release out and volatilize as time goes on [11]. Some good solutions could be expressed by information elements. To reduce errors in the late stage, ants will establish new solutions to include the good solutions. In multi-objective ant colony optimization algorithm, updating of information elements includes local updating of information elements and global updating of information elements. When a virtual machine was allocated to a host, ants would reduce the grade of information elements tracking from the virtual machine to the host. The computational formula was as follows.

$$\omega_{m,n}(u)=(1-\eta_a)\omega_{m,n}(u-1)+\eta_a\bullet\omega_0, \tag{8}$$

where ω_0 stands for the grade of information elements in the initial stage and η_a stands for the volatilization parameter of local information elements.

After all ants had solutions, global information elements would be updated. For each solution G, global information elements were updated according to the following equation.

$$\omega_{m,n}(u)=(1-\eta_a)\omega_{m,n}(u-1)\frac{\eta_b \bullet \beta}{K'(G)+M(G)}, \quad (9)$$

where $\beta=\frac{LA}{u-LI_s+B}$ and η_b stands for the volatilization coefficient during updating of global information concentration. When global non-dominated solutions which constituted the optimal solution set were in another set, the iteration solution was uncontrollable, then the solution was allocated to the set of non-dominated solutions; in the process of allocation, information elements increased. LA stands for the number of ants and LI_s stands for the iterative times of non-dominated solutions. B was accommodation coefficient, and the contribution of solutions in extra set to information elements could be controlled through B.

IV. Experimental Results and Analysis

Cloudsim software [14] was used to test the multi-objective ant colony optimization algorithm and verify the feasibility of the algorithm.

Cloudsim software was used to create a virtual experimental environment. The scheme of virtual machine resource allocation based on multi-objective ant colony optimization algorithm was imported to the experimental environment.

A. Setting of experimental parameters

Experimental parameters were set before experiment.

One hundred physical nodes were set. The CPU computational ability of a physical host was also set. The host had 8 G memory and was installed with 1TB mechanical hard disk. There were 200 virtual machines. The CPU computational ability of virtual machines was 200, 400, 600 and 800 respectively; the CPU calculation frequency of virtual machines was 0.2 GH, 0.4 GH, 0.6 GH and 0.8 GH respectively. The internal memory and storage space of virtual machines were 1 GB and 400 GB respectively. When the CPU of the physical host was totally occupied, power consumption was 200 W; when the utilization rate was 0%, power consumption was 150 W. Two hundred tasks proceeded simultaneously. The inspiring factor of information elements was 1, the inspiring factor of visibility was 4, the volatility coefficient of information elements was 0.3, the iterative times was 25, the number of ants was 40, and the strength of initial information elements was 5.

B. Experimental results and analysis

Based on the above experimental parameters, simulation experiment was carried out on first adaptation algorithm, least residue algorithm, conventional ant colony algorithm and multi-objective ant colony optimization algorithm. During experiment, the four algorithms were expressed as A, B, C and D. Then the algorithms were compared in the aspects of SLA violation rate [15], resource balance and placement superiority to obtain the optimal algorithm. The experimental results are shown in Fig. 3.

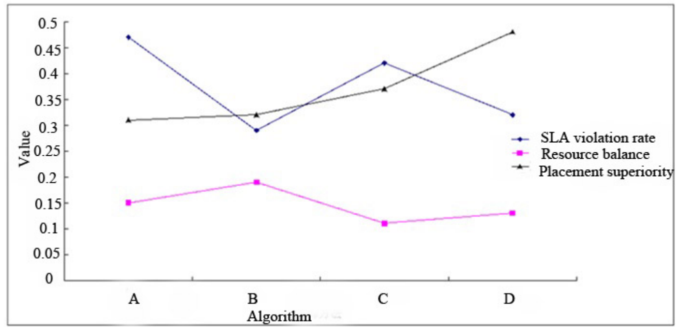

Fig. 3 The SLA violation rate, resource balance and placement superiority of the four algorithms

As shown in Figure 3, the placement superiority of algorithm A was the lowest, suggesting the algorithm had certain randomness; but the other two indicators of algorithm A were moderate. Algorithm B had the highest resource balance value, suggesting it had the most residual resources; as a result, virtual machines were placed in more hosts, which decreased the SLA violation rate. Higher SLA violation rate indicated poorer service performance and lower reliability. Algorithm C had relatively low SLA violation rate, suggesting better service performance and reliability; moreover, its energy consumption was also low. But algorithm D which was the improved algorithm of algorithm C had higher resource balance value and lower SLA violation rate, suggesting algorithm D had less energy consumption and better service performance; moreover, the placement superiority of algorithm D was the highest. It was concluded that the SLA violation rate and resource balance of first fit algorithm were moderate, least residue algorithm had large power consumption but high resource balance and low SLA violation rate. The conventional ant colony algorithm had excellent service quality and low energy consumption. Multi-objective ant colony optimization algorithm had better service quality and lower energy consumption compared to the conventional ant colony algorithm. To sum up, multi-objective ant colony optimization algorithm can ensure a relatively high degree of resource balance and less energy consumption on the premise of high placement superiority. Therefore, multi-objective ant colony optimization algorithm had a favorable adaptability in OpenStack virtual machine resource scheduling.

V. Conclusion

With the high-speed development of cloud computing and big data technology, Cloud computation has been one of the hottest information technologies. OpenStack cloud computation platform as the most potential cloud platform has attracted many attentions; however, the problem of resource scheduling severely limits the development of the platform. For OpenStack, if a reasonable resource allocation rule can be used, the distribution and utilization of the system can be obtained in real time, and the performance of the system can be adjusted in time. Moreover selecting the best scheme for machines with different utilization rates after considering the service quality, power consumption and resource consumption of machines can effectively solve the problem of high power consumption and low efficiency of physical machine and prevent the system service quality from being reduced because of the high load.

In order to achieve the above objectives, this study analyzed the basic features, technical architecture and core technology of the OpenStack cloud platform. Aiming at the problems appearing in the data migration, the multi-objective ant colony optimization algorithm was proposed. The improved algorithm performed better in solving the resource scheduling problem of OpenStack virtual machine and allowed the OpenStack virtual machine to meet the needs of users more quickly. Finally the simulation experiments were carried out on first fit algorithm, least residue algorithm, the conventional ant colony algorithm and the multi-objective ant colony optimization algorithm. The simulation results demonstrated that first fit algorithm had high randomness because it aimed at selecting the first physical host which satisfied requirements and moderate SLA violation rate and resource balance; least residue algorithm was featured by high energy consumption, high resource balance and low SLA violation rate; the conventional ant colony algorithm with positive feedback mechanism had low SLA violation rate and resource balance, i.e. high service quality and low energy consumption; the multi-objective ant colony optimization algorithm which was the improved version of the conventional ant colony algorithm had lower SLA violation rate and resource balance compared to the conventional ant colony algorithm. The multi-objective ant colony optimization algorithm had the highest placement superiority. It balanced SLA, resource balance and energy consumption and achieved high service quality and resource balance and low energy consumption, indicating a high performance. The multi-objective ant colony optimization algorithm has some shortcomings because of the limited skills of the designer, which remains to be further studied in the future. For example, only the balance between CPU, internal storage and broadband resources were considered in resource balance. Moreover double threshold technique was used in the design of algorithm, but a reliable model for determining thresholds was not given, and the selection of thresholds could directly affect the accuracy of the algorithm. One of the future research directions is to give the model for determining threshold. The third shortcoming was that past resource information in virtual machines needed to be collected, but they could be only used for determining whether there was migration. Therefore another research direction in the future can be construction of a model which can predict future resources based on the previous resource information. A key research direction concerning cloud computing in the future can be providing more computation resources in unit energy consumption.

VI. Acknowledgement

This study was supported by a major social science project of the Tianjin Municipal Education Commission: Research on the development strategy of education informatization in Tianjin "2030 of China's education modernization" (Project number:2017JWZD28).

References

[1] W. T. Tsai, X. Sun and J. Balasooriya, “Service-Oriented Cloud Computing Architecture,” in *2010 Seventh International Conference on Information Technology: New Generations*, pp. 684-689.

[2] C. C. Chen, P. L. Sun, C. T. Yang, J. C. Liu, S. T. Chen and Z. Y. Wan, “Implementation of a Cloud Energy Saving System with Virtual Machine Dynamic Resource Allocation Method Based on OpenStack,” in *Seventh International Symposium on Parallel Architectures, Algorithms and Programming*, pp. 190-196.

[3] G. Yang and W. Zhang, “Research of optimized resource allocation strategy based on OpenStack,” in *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pp. 365-368.

[4] M. Denis, J. Castro Leon, E. Ormancey and P. Tedesco, “Identity federation in OpenStack - an introduction to hybrid clouds,” *Journal of Physics Conference Series*, 2015, pp. 022015.

[5] V. Shamugam, I. Murray, J. A. Leong and A. S. Sidhu, “Software Defined Networking challenges and future direction: A case study of implementing SDN features on OpenStack private cloud,” *Materials Science and Engineering Conference Series*, 2016, pp. 012003.

[6] T. Rosado and J. Bernardino, “An overview of openstack architecture,” in *Proceedings of the 18th International Database Engineering & Applications Symposium*, 2014, pp. 366-367.

[7] S. G. Kang and K. W. Lee, “Testing Implementation of Remote Sensing Image Analysis Processing Service on OpenStack of Open Source Cloud Platform,” *Journal of the Korean Association of Geographic Information Studies*, vol. 16, no. 4, pp. 141-152, 2013.

[8] S Chen, J He and H Liu, “Realization and simulation of parallel ant colony algorithm to solve WTA problem,” in *2012 International Conference on Systems and Informatics*, pp. 2458-2461.

[9] Z. Liu, T. Liu and X. Gao, “An Improved Ant Colony Optimization Algorithm Based on Pheromone Backtracking,” in *2011 IEEE International Conference on Computational Science and Engineering*, pp. 658-661.

[10] Z. Zhang, N. Zhang and Z. Feng, “Multi-satellite control resource scheduling based on ant colony optimization,” *Expert Systems with Applications*, vol. 41, no. 6, pp. 2816-2823, 2014.

[11] K. Krynicki, M. E. Houle and J. Jaen, “An efficient ant colony optimization strategy for the resolution of multi-class queries,” *Knowledge-Based Systems*, vol. 105, no. C, pp. 96-106, 2016.

[12] Z. Xu, L. Xiao, W. Zhan, X. C. Yue, L. Ruan and R. Liu, “VM resource constraints based on hierarchy for OpenStack,” in *2016 2nd International Conference on Cloud Computing and Internet of Things*, pp. 30-34.

[13] Y. Yamato, Y. Nishizawa, M. Muroi and K. Tanaka, “Development of resource management server for production IaaS services based on OpenStack,” *Journal of Information Processing*, vol. 23, no. 1, pp. 58-66, 2015.

[14] T. Vondra, J. Šedivý and J M Castro, “Modifying CloudSim to accurately simulate interactive services for cloud autoscaling,” *Concurrency & Computation Practice & Experience*, vol. 29, no. 1 and 2, pp. e3983, 2017.

[15] S. Singh, I. Chana, and R. Buyya, “STAR: SLA-aware Autonomic Management of Cloud Resources,” *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1-1, 2017.

[16] Y. Yamato, Y. Nishizawa, S. Nagao, and K. Sato, “Fast and reliable restoration method of virtual resources on OpenStack,” *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1-1, 2015.

[17] H. T. Chang and S. Y. Wang, “Using SDN technology to mitigate congestion in the OpenStack data center network,” *IEEE International Conference on Communications*, pp. 401-406, 2015.

[18] Y. W. Kwon and H. Bang, “The finite element method using MATLAB (2nd ed.),” *CRC Press*, 2015.

[19] A. Ahmed and M. Ibrahim, "Analysis of Energy Saving Approaches in Cloud Computing using Ant Colony and First Fit Algorithms," *International Journal of Advanced Computer Science & Applications*, vol. 8, no. 12, pp. 248-254, 2017.

[20] J. K. Davidson, S. B. Savaliya, Y. He, J. J. Shah, "Methods of robotics and the pseudoinverse to obtain the least-squares fit of measured points on line-profiles," *Desogm for Manufacturing and Life Cycle Conference*, pp. 347-358, 2015.

[21] X. L. Zhang, W. Chen, B. J. Wang, X. F. Chen, "Intelligent fault diagnosis of rotating machinery using support vector machine with ant colony algorithm for synchronous feature selection and parameter optimization," *Neurocomputing*, vol. 167, no. C, pp. 260-279, 2015.